Anomaly detection in gravitational waves data using convolutional autoencoders

MACHINE
LEARNING
Science and Technology

CrossMark

OPEN ACCESS

RECEIVED
28 November 2020

REVISED
1 March 2021

ACCEPTED FOR PUBLICATION
31 March 2021

PUBLISHED
19 July 2021

PAPER

Anomaly detection in gravitational waves data using convolutional autoencoders

Filip Morawski[1,*], **Michał Bejger**[1], **Elena Cuoco**[2,3,4] **and Luigia Petre**[5]

[1] Nicolaus Copernicus Astronomical Center, Polish Academy of Sciences, Bartycka 18, 00-716 Warsaw, Poland
[2] European Gravitational Observatory (EGO), I-56021 Cascina, Pisa, Italy
[3] Scuola Normale Superiore, Piazza dei Cavalieri 7, I-56126 Pisa, Italy
[4] INFN, Sezione di Pisa, Largo Bruno Pontecorvo, 3, I-56127 Pisa, Italy
[5] Department of Computer Science, Faculty of Science and Engineering, Åbo Akademi University, Tuomiokirkontori 3, 20500 Turku, Finland
[*] Author to whom any correspondence should be addressed.

E-mail: fmorawski@camk.edu.pl

Keywords: gravitational waves, machine learning, autoencoders, convolutional neural networks

Abstract

As of this moment, 50 gravitational wave (GW) detections have been announced, thanks to the observational efforts of the LIGO-Virgo collaboration, working with the Advanced LIGO and the Advanced Virgo interferometers. The detection of signals is complicated by the noise-dominated nature of the data. Conventional approaches in GW detection procedures require either precise knowledge of the GW waveform in the context of matched filtering searches or coincident analysis of data from multiple detectors. Furthermore, the analysis is prone to contamination by instrumental or environmental artifacts called glitches which either mimic astrophysical signals or reduce the overall quality of data. In this paper, we propose an alternative generic method of studying GW data based on detecting anomalies. The anomalies we study are transient signals, different from the slow non-stationary noise of the detector. The anomalies presented in the manuscript are mostly based on the GW emitted by the mergers of binary black hole systems. However, the presented study of anomalies is not limited only to GW alone, but also includes glitches occurring in the real LIGO/Virgo dataset available at the Gravitational Waves Open Science Center. To search for anomalies we employ deep learning algorithms, namely convolutional autoencoders, which are trained on both simulated and real detector data. We demonstrate the capabilities of our deep learning implementation in the reconstruction of injected signals. We study the influence of the GW strength, defined in terms of matched filter signal-to-noise ratio, on the detection of anomalies. Moreover, we present the application of our method for the localization in time of anomalies in the studied time-series data. We validate the results of anomaly searches on real data containing confirmed gravitational wave detections; we thus prove the generalization capabilities of our method, towards detecting GWs unknown to our deep learning models during training.

1. Introduction

The first gravitational wave (GW) detection on 14 September 2015 [1] inaugurated a new era in astrophysics. The joint observational effort of the LIGO and Virgo Collaborations, working with the Advanced LIGO (aLIGO) [2] and the Advanced Virgo (aVirgo) [3] interferometers in a global network has provided 50 GW candidate signal detections, until the suspension of LIGO-Virgo observing run O3 in Spring 2020 [4–7]. Such an impressive number of statistically-significant signal candidates allows for the verification of many theoretical models, describing various sources of GW radiation, like binary systems of black holes (BHs) and neutron stars (NSs), as well as the very nature of gravity [8]. As a result of continuous work to improve their sensitivity, the aLIGO and the aVirgo interferometers will soon probe a much larger volume of space and expand the capability of discovering new GW sources. According to theoretical predictions, tens of BH mergers and a few NS mergers will be soon routinely registered every year [9]. Such a large number of events will deliver extraordinary information about the nature of those objects, phenomena, as well as space-time itself.

What can be observed through the GW window strongly depends on the fidelity of the data analysis methods. The GW data is always noise dominated: astrophysical signals are buried deep into the detectors' noise, caused by various sources. Some of them are associated with the environment [8], e.g. seismic and environmental activity; others originate from the detector itself [9], such as the thermal fluctuations of the mirrors and the laser beam photon shot noise [10]—time-dependent fluctuations in the laser interacting with the mirrors of the interferometer. In order to reveal the hidden GWs, the traditional approach is based on matched filtering algorithms [11]. The examples of existing pipelines utilising a matched filtering approach are PyCBC [12] and GstLAL [13]. Assuming the model gravitational waveform is known, which is not always the case for all astrophysical sources, the algorithm scans the data to match an optimal template using a template bank. Matched filtering is computationally expensive and in the case of large template banks requires a lot of computational resources. As an optimal method of filtering in stationary Gaussian noise, it is prone to contamination by non-stationary instrumental artifacts, the so-called glitches, which either mimic GW signals or reduce the quality of the collected data. Existing alternatives to matched filtering methods are based on unmodeled searches of GW burst signals. Such GWs are of short duration with an unknown or partially modelled waveform morphology related to complicated or unknown astrophysics. These signals are searched by measuring an excessive power in the time-frequency domain that occurs coherently between multiple detectors. An example of GW burst pipeline is coherent WaveBurst (cWB) [14, 15]. The sensitivity of this method is affected by short duration glitches that may occur coincident in time between multiple detectors. Versatile and rapid pipelines are needed to deal with non-stationary noise (and instrumental glitches) and to perform preliminary analysis of large amounts of data from multiple detectors.

Deep learning (DL) [16] fits this role perfectly. DL has commenced a new era of machine learning (ML), a field of computer science based on specially-designed algorithms that can generalize ('learn') from examples in order to solve problems and make predictions, without the need of being explicitly programmed [17]. DL algorithms, based on the concept of neural networks—models of neuron connectivity in the brain—are able to analyse different representations of data with varied dimensionality, like images (spectrograms, Q-transforms, Wavelet-transform) or time series. Moreover, they can quickly process large amounts of data—a requirement for the real-time (low latency) analysis for the GW interferometers.

The purpose of this work is to propose a generic, model-independent data analysis method that searches for anomalies in signals recorded by the GW detectors, based on performant DL models. The innovation of the proposed method is that our DL model 'learns' the features of the noise and detects if anomalies occur in the detected signal. We define an anomaly as an extraordinary, sparse transient data feature, outstanding with respect to the 'normal' background noise of the detector. The anomaly signal could therefore either be represented as a GW or an instrumental glitch. In particular, the GWs studied in the presented work are based on the signals emitted by the binary BH systems (BBH).

The term 'anomaly' is a well defined concept in statistics and data analysis (hence also in machine learning), sometimes also referred to as 'outlier'. The DL methods we apply are perfectly fitted for their detection and analysis role here, being especially suited to detect, compress, and reconstruct non-linearities in the input signal data. The analysis assessing the capabilities of the method is performed with the BBH signals, injected (added) to both simulated and real detector data. Once trained and tested on injections, the DL models are validated on the confirmed GW detections—real astrophysical signals registered by the detectors. Our results provide a proof-of-concept for the advantages of using DL methods in the GW data analysis, namely the processing speed and ability to capture complicated non-linear relationships in the data. These features enable our method to be potentially used as an event-trigger-generator (ETG). In this context, the advantage of our method is the computational speed—ML algorithms once trained are extremely efficient in the processing of data, in particular when used on computers equipped with graphics processing units (GPUs). By searching for anomalies in the gravitational waves' data via ML algorithms, our method could support the currently existing ETGs such as Omicron [18] and Q-transform based Omega [19].

In GW astronomy, while DL is being actively researched, it is still quite a novel method. Therefore this research fits very well in the early adoption scheme of the modern state-of-the-art development in the field, knowledge of which will become indispensable in the near future. In the following we mention a few interesting test cases. George and Huerta [20] developed the deep filtering method for signal processing, based on a system of two deep convolutional neural networks, designed to detect and estimate parameters of compact binary coalescence signal in highly noisy time-series data streams. The same authors have been involved in a group working on denoising gravitational waves with autoencoders [21]. Dreissigacker *et al* [22] have been using DL as a search method for continuous GWs emitted by spinning neutron stars. Similar work has been conducted by Morawski *et al* [23] on the application of convolutional neural networks for classifying continuous GW signal candidates. Furthermore, DL has been used by Beheshtipour and Papa [24] for clustering continuous GW candidates. DL has also been successfully used in the classification of glitches by Razzano and Cuoco [25]. Finally DL has been used in searches for GW emitted by core-collapse supernova explosions by Iess *et al* [26].

Within GW astrophysics, application of DL in anomaly detection is a fairly new concept (see however a recent application in [27]). In different fields of astrophysics the ML anomaly detection has already been successfully implemented [28, 29] (see [30] for a recent general overview of ML in astronomy). Outside astronomy, anomaly detection has also been proposed in the search for signatures of new physics in the Large Hadron Collider data [31], where it was demonstrated to discover a specific class of highly-energetic particle jets, without the prior knowledge of their specific features.

The outline of this work is as follows. In section 2 we briefly discuss the DL architecture applied in our work and in section 3 we describe our data generation procedures. In section 4 we explain the results of the anomaly detection studies performed on both simulated and real data. Finally, in section 5 we further discuss our results and draw some conclusions.

2. Deep Learning algorithms

In this paper we employ a combination of two deep learning methods for distinguishing GWs from noise signals. We first briefly describe these learning methods and then we explain how we applied them to our problem.

2.1. The convolutional neural network and autoencoder

A *convolutional neural network* [16] (CNN) is a deep, feed-forward artificial neural network (processes the information one-way, from the input to the output), designed for processing structured arrays of data, e.g. for classifying images. The core feature of a CNN is the convolution operator, that differentiates them from regular (linear) neural networks employing simple matrix multiplication. The convolution operation envisions the input structured array (say, of dimension $m \times n$) as a sequence of overlapping elements of dimension, say, $p \times p$, often with $p < min(m,n)$. The convolution operator inner-multiplies each such element with a so-called kernel (or filter), of the same dimension ($p \times p$), that 'slides' over each element of the original array. Each $p \times p$ element of the input array is thus replaced by the result of the convolution operation (a number), and we thus have dimensionality reduction. This somewhat simplified convolution concept is illustrated in figure 1.

Multiple layers are typically sequentially used, each with varying numbers and types of kernels. The kernels are chosen so that the network learns specific features in one convolution (e.g. edges, corners, etc). Convolution layers are alternated with other specialised layers, all further reducing the dimensionality, until a final activation function maps the last layer to a vector whose elements correspond to the desired options (classes) for classifying the input structured array. Each class in this vector is usually a probability, so that the class with the highest value is indicated as the recognised class.

The overlapping of the input array elements together with the sliding kernel, as well as the sequencing of applying convolution step-wise simulates the structure and operation of the visual human cortex that processes incoming images in a series of layers, by identifying progressively more complex features. CNNs have been shown to work extremely well at picking up patterns in the input structured arrays, for instance in images, and thus have rapidly become the state-of-the-art in image classification and computer vision.

In our approach we use all the advantages brought by CNNs, but we further embed them into an *autoencoder* (AE) architecture [32]. An AE [16] is a special type of deep artificial neural network that step-wise encodes and compresses the input and then it (re)constructs an output based only on the most compressed encoding, called the hidden layer, latent representation, or bottleneck. The main AE hypothesis is that some structure exists in the input data, for instance some form of correlation between the input features; such a structure can be and is learnt by an AE and consequently leveraged when forcing the input through the latent layer. (If the input features are independent of each other, then the compression and subsequent reconstruction are very difficult.)

An ideal AE is sensitive enough to the inputs to accurately build a reconstruction and insensitive enough to the same inputs so that the model does not memorize/overfit the training data. Differently formulated, this forces the model to maintain in the latent representation only the variations in data needed for reconstruction, without holding on to redundancies within the input. There are different types of AE, but in our work we use the undercomplete AE, where the dimension of the latent representation is strictly smaller than the input dimension. In this way we take care of avoiding overfitting, since the model will not be able to copy the input to the output.

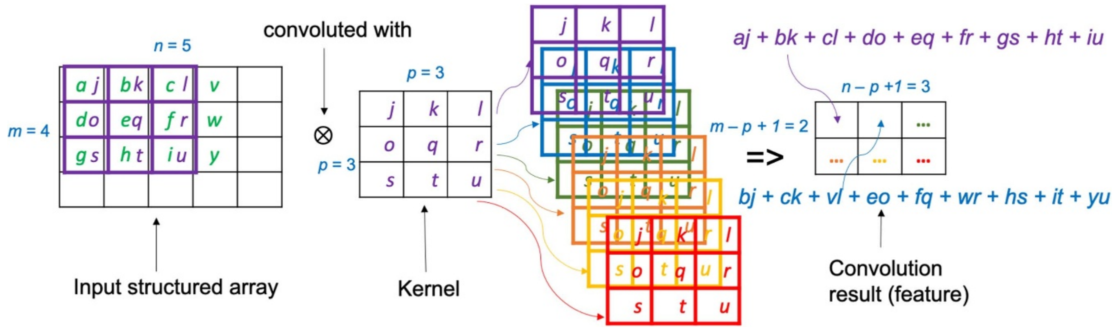

Figure 1. The 2-dimensional 4 × 5 array on the left is seen as six overlapping 3 × 3 arrays; each of these smaller arrays is inner-multiplied with the 3 × 3 kernel, resulting in the 2 × 3 array on the right.

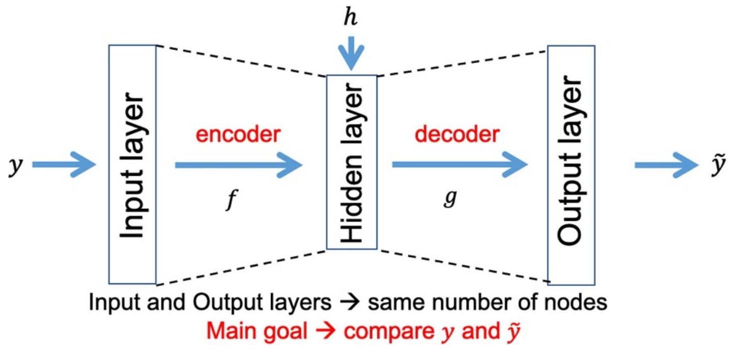
h
y
Input layer
encoder
f
Hidden layer
decoder
g
Output layer
ỹ
Input and Output layers → same number of nodes
Main goal → compare y and ỹ

Figure 2. An AE with one hidden layer.

The simple AE in figure 2 (with only one hidden layer) works by first encoding the input $y \in \mathbb{R}^d$ to the element $h \in \mathbb{R}^p$, where $p < d$. Assuming we have the encoder activation function $f : \mathbb{R}^d \rightarrow \mathbb{R}^p$, then $h = f(Wy + b)$, where f can be sigmoid, `ReLU`, or something else, W is a weight matrix and b a bias factor (initialised randomly and updated iteratively during training). The decoder activation function $g : \mathbb{R}^p \rightarrow \mathbb{R}^d$ will map h to $\tilde{y} = g(\tilde{W}h + \tilde{b})$, where the activation function g, the weight matrix $\tilde{W}$ and the bias factor $\tilde{b}$ may be unrelated to f, W, and b. If only `ReLU` is used for activation and we have only one hidden layer, then we have a linear AE; if we have more hidden layers, or non-linear activation(s), then the AE becomes non-linear, which is better at detecting abstract features. Thus, in general, the encoder and decoder are proper neural networks in themselves, not simply activation functions.

The training of the AE works by attempting to minimize the reconstruction loss, most often using the mean squared error (MSE) formula [16]

$$\mathcal{L}(y, \tilde{y}) = \|y - \tilde{y}\|^2 = \left\|y - g(\tilde{W}f(Wy + b) + \tilde{b})\right\|^2 . \tag{1}$$

The training of the network works by updating the parameters W, b, $\tilde{W}$, $\tilde{b}$ until $\mathcal{L}(y, \tilde{y})$ is sufficiently small and further training does not decrease it anymore—in that case the network has converged. There are numerous algorithms for updating the parameters, but here we use the ADAM algorithm (adaptive moment estimation) [33], as it adapts the learning rate during training and has been empirically shown superior to other methods for large datasets, large number of parameters, as well as non-stationary input. The learning rate (training 'step' of updating) together with the batch size (number of training examples that use the same learning rate), the update method (also called as optimizer), the number and size of the layers and the loss function are the hyperparameters of the AE.

Since the AE is reducing dimensionality for encoding, it has often been used for only that—for instance, for feature learning. However, when compared to other dimensionality reduction techniques, such as principal component analysis (PCA), AE appears as a powerful generalization, since it is able to learn non-linear relationships in input data. While PCA attempts to discover a lower dimensional hyperplane describing the original data, AE is capable of learning non-linear manifolds of the least possible size, as illustrated in figure 3. Essentially, the AE learns a vector field for mapping input data towards lower

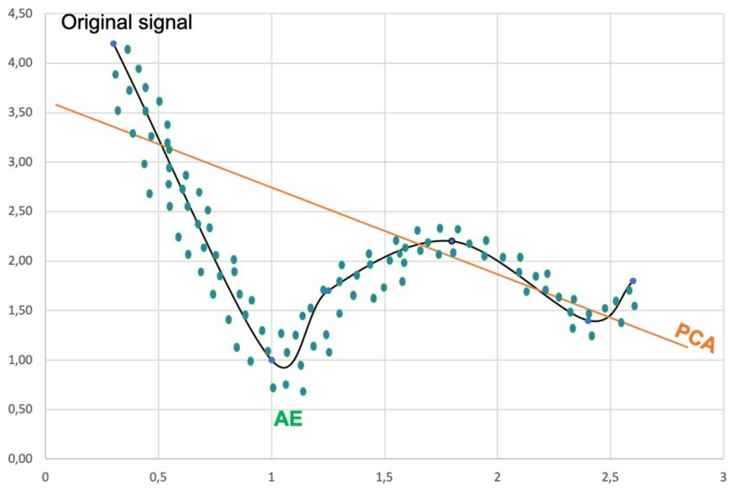

Figure 3. Example of the manifold concept. Nonlinear vs linear dimension reduction.

dimensional manifolds, that describe the high density region where input data concentrates. If the manifold accurately describes input data, then the AE has effectively learnt the input data.

2.2. Applying the CNN AE

Although CNNs were designed for the analysis of 2D data (i.e. images) [16], here we apply them for a simpler, 1D implementation, as our analysed signals are time series. The CNN AE is designed to learn two things. In case of input data instances containing only detector background noise (no anomalies), the AE is trained to reconstruct the noise as closely as possible. However, in case of instances containing an anomalous signal (GW or glitch in case of real data), the AE is trained to disregard the anomaly and reconstruct the data as if the signal was not present. By comparing the input to the output reconstructed by the AE, the anomaly present in the instance of data time series is recovered and further studied. Using the AE network is thus instrumental, since it will reconstruct only the signals it has been trained for, in our case the detector noise, and disregard anything else in its reconstruction, in our case the GW or the glitch, as noise.

Depending on the presence of anomalies in the input data, the loss value $\mathcal{L}(y,\tilde{y})$ is expected to vary. The $\mathcal{L}(y,\tilde{y})$ computed for an 'anomalous' input reaches higher values than in case of 'anomaly-free' input, since the difference between y (noise or noise with anomaly) and $\tilde{y}$ (only noise) is larger. The difference between these signals is proportional to the amplitude of the anomaly. As a result, the AE trained on data containing stronger anomalies is expected to converge during training towards higher $\mathcal{L}(y,\tilde{y})$.

The final architecture[6] used in section 4 was chosen based on empirical tests on the data. We tested architectures ranging from one to eight hidden layers. The final layout of the architecture is presented in figure 4. The chosen architecture reaching the minimum value for $\mathcal{L}(y,\tilde{y})$ for the fixed training data set was the network containing three hidden convolutional layers: an encoding layer, a decoding layer, and a latent representation layer in between them, with 256, 128, and 256 neurons, respectively. The kernel size was fixed for all layers to 3 × 1. All but the final output layer use the `ReLU` as the activation function, whereas the final layer reconstructing the initial signal uses a sigmoid activation function. The other hyperparameters used for training were the ADAM optimizer [33] with learning rate of 0.0005 and batch size of 32.

In the following section, we detail our datasets used for training, validation and testing.

3. Training data sets and data flow

We prepared two kinds of training datasets: a simplified one by means of simulated detector strain time series (based on the colored normal distribution of noise), denoted DataSet 1 (DS1), and a realistic one based on the real LIGO-Virgo O2 observing run [39], publicly available at the Gravitational Waves Open Science Center (GWOSC) [40] and denoted DataSet 2 (DS2). In both cases we use the same general data flow presented in figure 5.

[6] We implement the algorithm in `Python` [34] using the `Keras/TensorFlow` library [35, 36] with the GPU support. The development was performed on the NVidia Quadro P6000 (sponsorship via the NVidia GPU seeding grant). The production runs were deployed on the Prometheus cluster (Academic Computer Centre CYFRONET AGH) equipped with Tesla K40 GPU nodes, running CUDA 10.0 [37] and the cuDNN 7.3.0 [38].

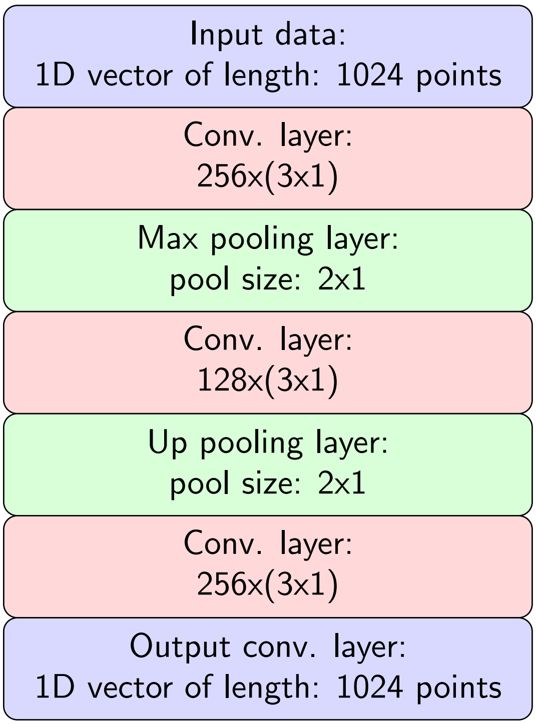

Figure 4. Diagram shows the networks' layer structure and architecture. The size below convolutional layers correspond to the *Number of filters* times *Kernels size* of every layer. The second dimension of all layers is equal to unity since the initial input data were 1D time-series vectors.

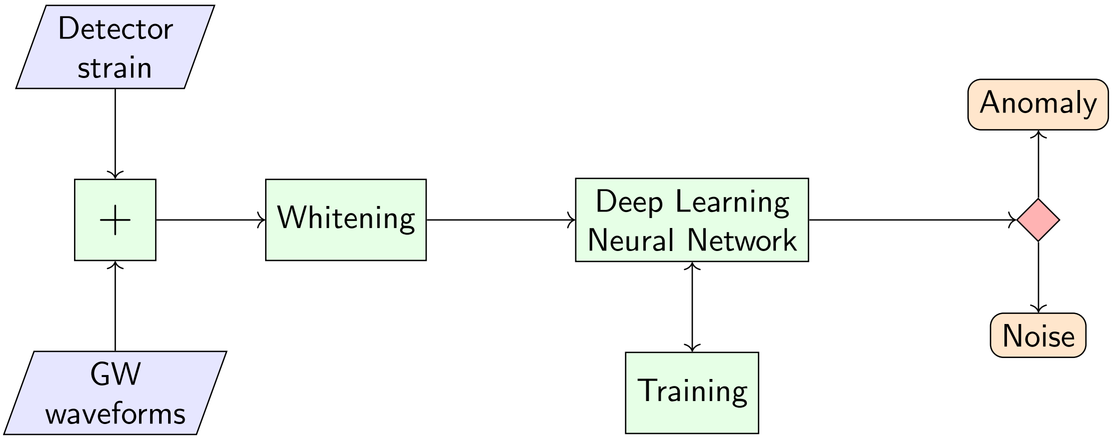

Figure 5. Data flow of the project. The first step consists of data generation. GW waveforms were injected into the detector strain, either simulated or real (e.g. public data from the GWOSC platform). Since the raw time-series signal varied with frequency, the whitening procedure was applied to simplify data for further training. The training of the DL algorithms aimed to recover as many injected anomalies as possible, while limiting the false positive rate of the pipeline (noise samples incorrectly classified as anomaly).

The whitening mentioned in the workflow diagram removes the contribution of the stationary detector noise and re-weights the sensitivity at different frequencies [41]. As a result, the amplitude spectral density of the data becomes uniform and GW signals buried in the data are easier to search for and compare with each other. The whitening filter was re-computed separately for DS1 and DS2 as well as for every interferometer to take into account the differences in the sensitivity. The whitening procedure used for the analysis of both studied datasets was conducted in the frequency domain following pyCBC Python library modules [42].

To simulate an astrophysical GW signal emitted by a BBH we used the IMRPhenomv4 waveform model [43], which includes the binary inspiral, merger of the components and the final BH ringdown. The component BH masses m_1 and m_2 of the waveform model were chosen to be compatible with the first detected GW150914 [1]. We selected m_1, m_2 based on the initial mass function (IMF) in ranges associated with uncertainties of mass estimation for GW150914: $m_1 : 32.5–40.3\ M_\odot$, $m_2 : 26.2–33.6\ M_\odot$. For the index

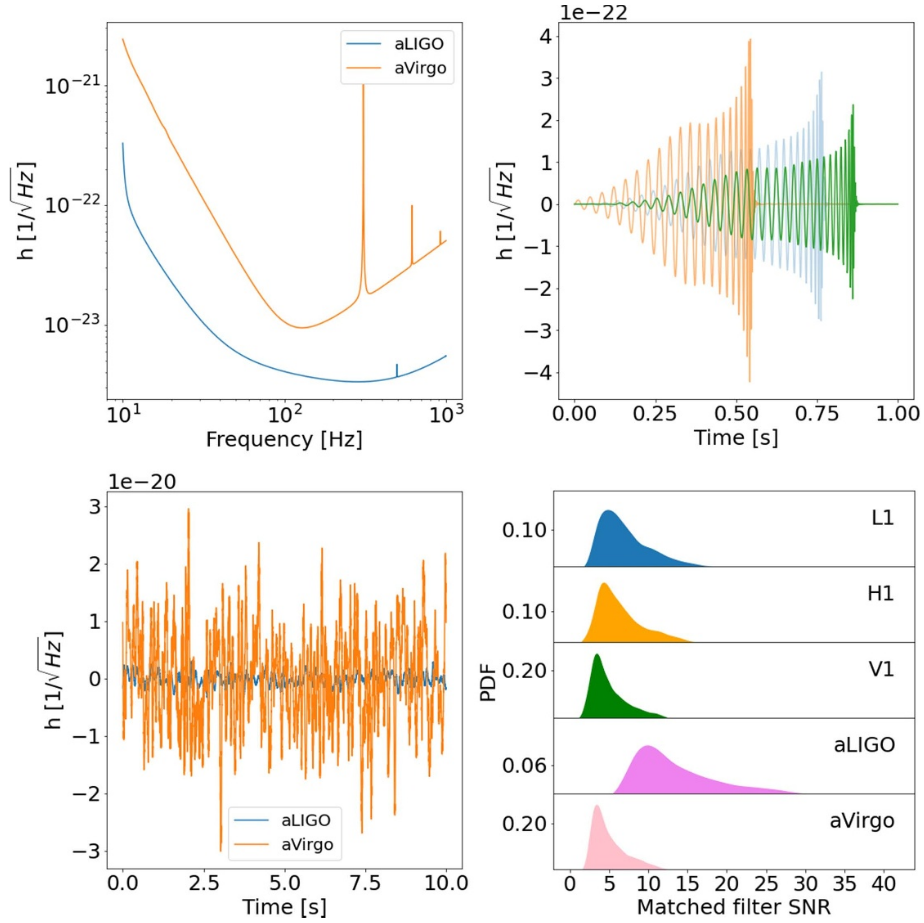

Figure 6. *Top left*: designed sensitivity of aVirgo and aLIGO interferometers. The detectors were expected to reach this level of sensitivity over broad range of frequencies after all planned upgrades. *Top right*: examples of generated BBH GW waveforms injected into the strain data as anomalies. The GW were generated for the following parameters: blue signal—$m1 = 29\ M\odot$, $m2 = 24\ M\odot$, distance = 440 Mpc; orange signal—$m1 = 33\ M\odot$, $m2 = 27\ M\odot$, distance = 380 Mpc; green signal—$m1 = 28\ M\odot$, $m2 = 23\ M\odot$, distance = 600 Mpc. *Bottom left*: examples of the simulated data using the above sensitivity curves. *Bottom right*: distributions of matched filter SNR of simulated GW injected into the real data for detectors: LIGO Hanford (blue), LIGO Livingston (orange) and Virgo (green) as well simulated data for: aLIGO (violet) and aVirgo (pink).

of the IMF power law, we chose the value of $\alpha = -2.35$ [44]. The luminosity distances were chosen uniformly from the range between 200 and 800 Mpc, to cover a realistic range of the matched filter signal-to-noise ratio (SNR)—from 4 to 40 varying for different interferometers as shown in the bottom right plot in figure 6. The position in the sky was chosen to be optimal for every detector in a given moment of time. The examples of a few simulated GW signals are presented on the top right plot in figure 6.

The DS1 dataset was created for the two assumed sensitivity curves of the GW detectors. Each curve described the level of the detector sensitivity with respect to the frequency in such a way that the generated strain was mimicking the realistic time series output. In our analysis we used the designed sensitivity for the aVirgo from O3 run (version without squeezing) [45, 46] and the aLIGO [47] interferometers. 'Designed' means that the interferometers were expected to reach this level of sensitivity after all the planned upgrades. Band-pass filtering was then applied to the generated noise to remove high frequency (above 1 kHz) and low frequency (below 30 Hz, corresponding to the seismic noise) components from the data, as current interferometers are not sensitive enough to detect GWs outside that frequency range. The data was then resampled from 4096 Hz to 1024 Hz. An example of the output time series from the DS1 is shown on the bottom left plot in figure 6. Prepared in advance GW signals were injected into the generated strain and subjected to the procedure of whitening.

The DS2 (realistic) dataset was created based on the publicly available LIGO-Virgo O2 observing run, using the data stored at the GWOSC platform [40] for three interferometers: LIGO Hanford, LIGO Livingston and Virgo described in the text with the respective abbreviations $H1$, $L1$ and $V1$. For each detector, we chose six hours of data to train the DL models. For $L1$, we took segments between 1187 270 656 and 1187 295 232 (in GPS time units); for $H1$, between 1174 958 080 and 1174 982 656; whereas for $V1$, between 1187 672 064 and 1187 696 640. We injected into the strain the same GW signals as for the simulated data. As a result, we obtained (for the same set of injections) three different distributions of matched filter SNR, since every detector had a different sensitivity. The distributions are illustrated in the bottom right plot in figure 6. For the comparison, we included additional SNR distributions of the simulated datasets. The obtained real data strain with injected anomalies was then subjected to the procedure of whitening and resampled from 4096 Hz to 1024 Hz.

In total, we generated five datasets (two simulated for aVirgo and aLIGO, and three based on real LIGO Livingston, LIGO Hanford and Virgo O2 data) which were further split into one second segments and divided for the training, validation and testing datasets (65%, 10% and 25% respectively). An additional test set, containing confirmed GW detections, has been created by using one hour of data around GPS time for each confirmed GW, whitened and resampled as described above. As confirmed GWs, we chose three BBH detections with the highest network SNR from O2 run: GW150914, GW170608 and GW170814 [39].

4. Results

The results presented below are split into subsections. The first presents the results of the anomaly searches on the simulated dataset containing injected GW. The second subsection covers searches of anomalies in the real data of Virgo and LIGO interferometers. The last subsection presents capabilities of anomaly searches on the real data containing confirmed detections of GW.

4.1. Anomaly searches on simulated data

The CNN-AE described in section 2 was first trained on the whitened, simulated data containing GWs. The convergence of the model was achieved after 100 epochs, during which the MSE loss function reached a value of around 6×10^{-5} for the aVirgo data and 10^{-4} for the aLIGO data. We extended the training for another 100 epochs to investigate the onset of overfitting. However, the overfitting did not appear and MSE fluctuated around the values mentioned above. The learning history of the AE, trained on both simulated datasets, with results for both the training and the validation sets is shown on the left plot in figure 7. For aVirgo, the same set of gravitational waveforms covered a SNR range of smaller values than for aLIGO, as a result of the worse detector sensitivity (see the top left plot in figure 6 for comparison). This, in turn, resulted in the convergence towards a lower MSE for the aVirgo dataset, since the difference between the 'anomalous' input and 'anomalous-free' reconstruction were smaller than in the case of aLIGO data (see section 2.2 for more details).

Furthermore, as described in section 2.2, the correctly trained CNN-AE reconstructed the pure detectors' noise, regardless of the anomalies presence in the input data. Then, by subtracting the input from the reconstructed output we aimed to recover the underlying signal. We computed the mentioned differences between the input initial data and the output AE reconstruction. Examples of the results are presented in figure 8, where blue, dashed lines in the background correspond to the expected results, whereas red lines correspond to values obtained with the AE. In the majority of cases, we were able to correctly reconstruct the GW waveforms for aLIGO data. The anomaly was significantly different from the surrounding noise, with its recovered part mostly associated with the merger part of the gravitational waveforms, together with a small contribution from the inspiral part. In contrast, for the aVirgo dataset, the reconstruction was significantly worse, often dominated by the surrounding noise. In rare cases, as presented on the right plot in figure 8, the merger part was reconstructed. In the appendix A we present the summary of the match between the injected and reconstructed waveforms using $\langle x_1|x_2\rangle$ metric performed in the time domain.

Since the AE was able to correctly reconstruct the majority of anomalies buried in the data (at least for the aLIGO dataset), we aimed then to define a metric allowing automatic anomaly detection. We chose the MSE as such a metric, computed between the input data and the AE output as detailed in section 2.2. Figure 9 shows histograms of MSE for the two different signal types present in the studied data: noise and injected GW. As expected, values of MSE for the noise were much smaller than for the latter case and close to zero. There was a range of MSE in which histograms of both types overlapped (denoted with the burgundy colour in figure 9). Nevertheless, the majority of noise with injected GW instances in the aLIGO dataset and almost half of these instances in the aVirgo dataset had values larger than the noise. Moreover, we added the detection threshold (defined in the following paragraph) to the histograms to stress out how many of initially injected GW were correctly detected as anomalies (hatched area in figure 9).

We defined the threshold for the anomaly detection using the relation between the false positive rate (FPR) and MSE. Then, by fixing FPR at a particular value, we set the detection threshold (DT) on the corresponding MSE. In the presented analysis, we fixed FPR at 5%, resulting in the following thresholds: $\mathrm{DT}_{\mathrm{simV}} = 1.6 \times 10^{-5}$ for aVirgo and $\mathrm{DT}_{\mathrm{simL}} = 3.1 \times 10^{-5}$ for aLIGO. The results of the anomaly searches at FPR = 5% are shown in table 1 in the form of confusion matrix. Additionally, the comparison in the anomaly detection efficiency for both interferometers is shown on the left plot in the figure 10 in the form of

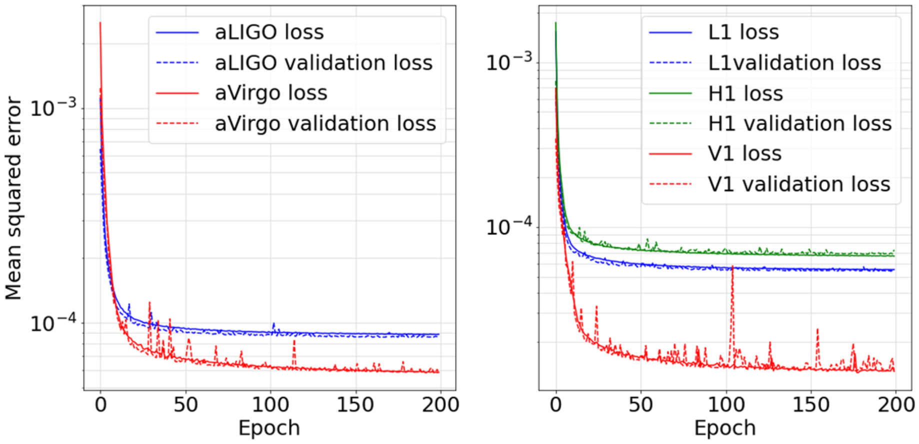

Figure 7. The learning history of the AE trained on the whitened, simulated aVirgo and aLIGO datasets (*left plot*) as well as on the whitened, real data for $L1$, $H1$ and $V1$ datasets (*right plot*). The convergence was achieved after 100 epochs. We trained for 200 epochs, to investigate the onset of overfitting: it did not appear.

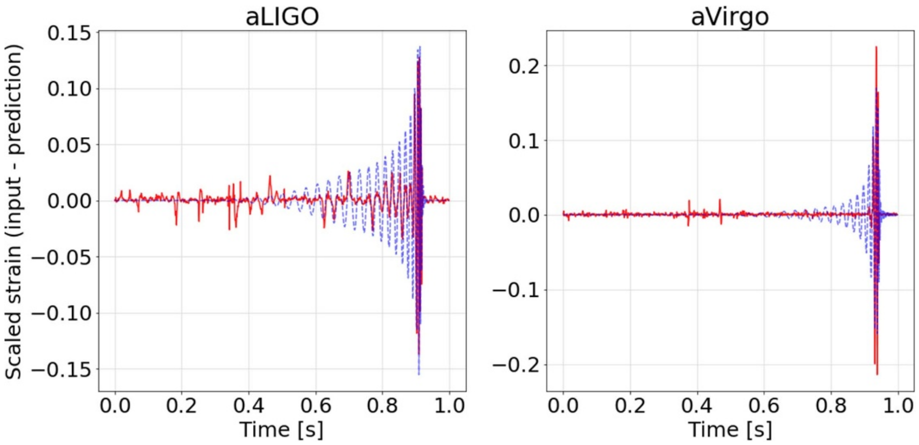

Figure 8. The examples of signal reconstruction using the CNN-AE on aLIGO (*left plot*) and aVirgo data (*right plot*). The plots (red curves) were generated after subtracting the input signal from the AE predictions. For comparison, we added the expected signal using blue, dashed curve (difference between the input signal and the ground-truth output signal).

receiver operating characteristic (ROC) curves. Over all ranges of FPR, the aLIGO detector achieved a significantly higher detection efficiency, or true positive rate (TPR).

Presented in table 1 values quantitatively represent the results from both panels in figure 9. Anomaly row relates to the hatched area from those panels. In the case of aLIGO dataset 96% of detected anomalies are correctly related to the injected GW with minor contamination of noise samples. In case of aVirgo 59% of samples characterized by low SNR (around 10 and less) samples did not exceed the DT and contributed to the non-anomalous group. Details presenting the relation between the SNR of the injected GW and MSE of the reconstructed waveform can be found in the appendix B.

4.2. Anomaly searches on real data

Later on, the AEs were trained on whitened, real data from the O2 observational run, collected for three existing interferometers: $V1$, $L1$ and $H1$ with injected BBH gravitational waveforms. The right plot in figure 7 presents the learning history of the AE trained on the dataset for every detector. As in the case of the simulated data, AE reached convergence after around 100 epochs. We again prolonged the training to investigate the onset of overfitting, which did not appear. Since the difference between the ‘anomalous’ input and the ‘anomalous-free’ reconstruction for $V1$ was the smallest among the considered datasets (as a result of the SNR smallest range), the AE during training converged towards the lowest MSE. Adequately, a smaller range of SNRs for the $H1$ dataset with respect to the $L1$ dataset resulted in the difference between the respective values of MSE (for $L1$ being higher, and for $H1$ being lower).

The ability of the AE to reconstruct the detectors’ noise was investigated as previously in the case of simulated data. The differences between the input strain and AE reconstructions were compared with the

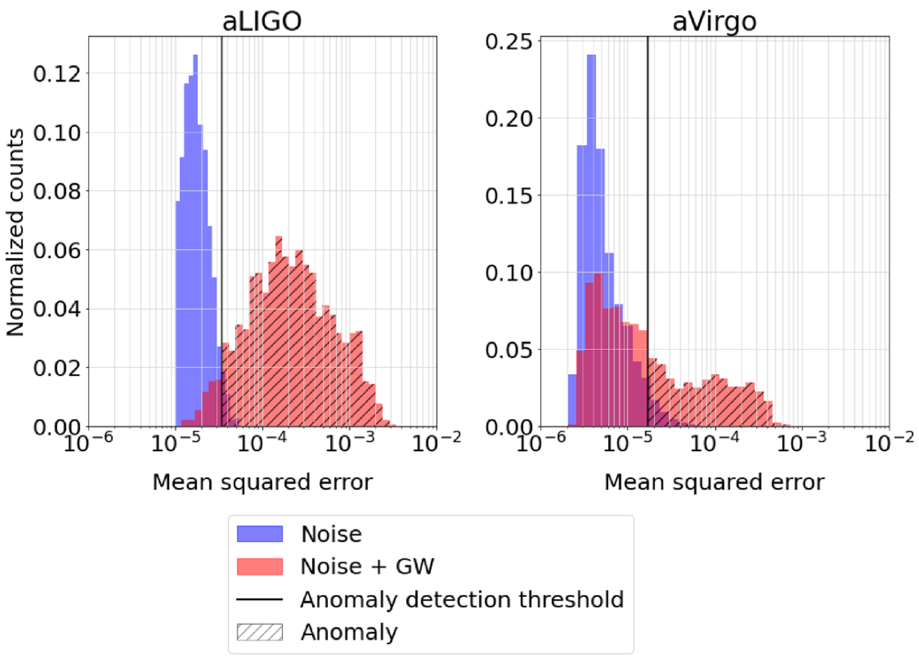

Figure 9. Distribution of MSE between the AE predictions and the input strain for two types of studied signals: noise (blue histogram) and injected GW (red histogram). In the broad range of MSE for aLIGO dataset the injected GW had significantly higher MSE than noise, allowing a definite distinction between these signal types. Additional black vertical line representing anomaly DT was added to emphasize amount of detected anomalies (hatched areas).

Table 1. Results of anomaly detection of CNN-AE at FPR = 5% for aLIGO and aVirgo dataset in the form of confusion matrix. Columns relate to the ground-truth values whereas rows to the predictions. For aLIGO dataset significant majority of detected anomalies corresponded to the data samples with injected GW. However in case of aVirgo more than a half of data samples with injected GW did not exceed the $\mathrm{DT}_{\mathrm{simV}}$ as a result of low SNR (see figure 6 for comparison between aLIGO and aVirgo GW SNR distributions).

	aLIGO		aVirgo	
	Injected GW	Noise	Injected GW	Noise
Anomaly	96%	5%	41%	5%
Non-anomaly	4%	95%	59%	95%

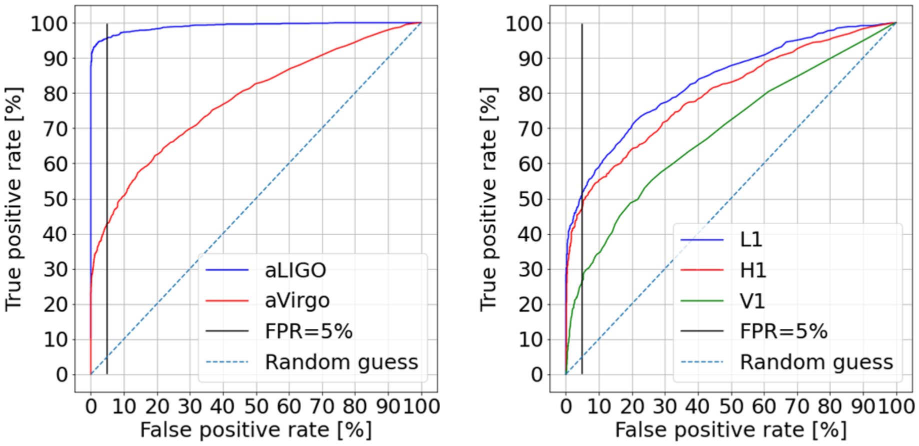

Figure 10. Receiver operating characteristic curves for simulated data (*left plot*) and real data (*right plot*). Black vertical line corresponds to the FPR = 5% chosen as the criterion for the anomaly detection threshold.

expected values. Examples of this comparison are presented in figure 11. The summary of the match between the injected and the reconstructed waveforms is presented in appendix A. The AE trained on $L1$ data achieved the best results, since the GW injected into the strain were extracted to the greatest extent in the merger part and partially in the inspiral part. On the other hand, the AE trained on the other datasets

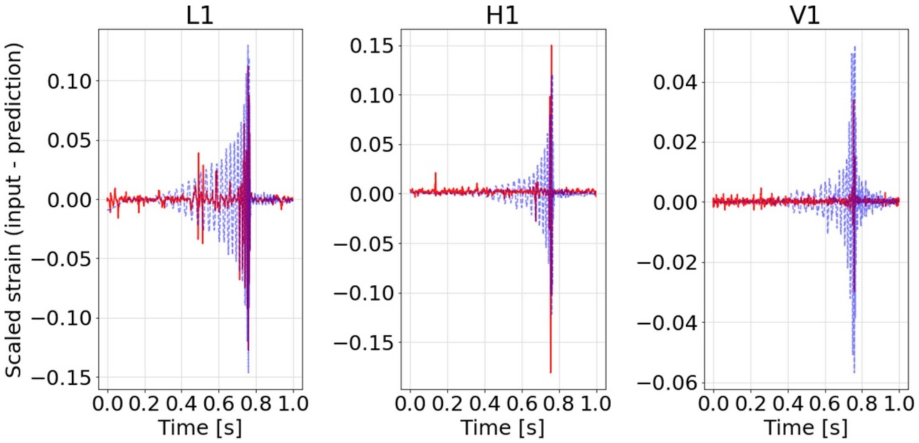

Figure 11. Three examples of GW reconstruction using AE. Presented plots were generated after subtracting from the AE predictions, input signal (red curve). For the comparison we added the expected signal using blue, dashed curve (difference between the input signal and the ground-truth output signal). From left to right: results for $L1$, $H1$ and $V1$ detectors.

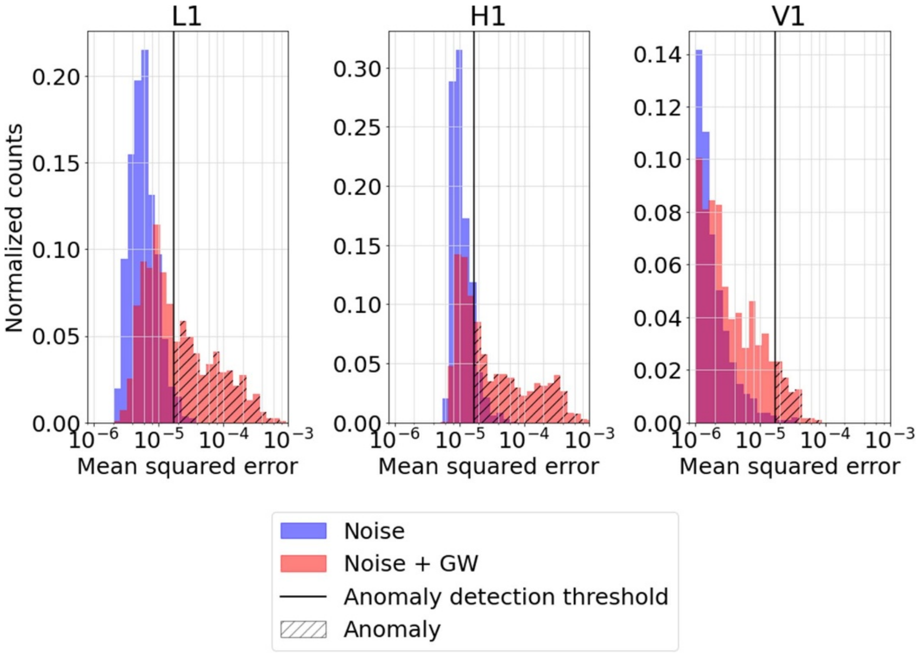

Figure 12. Distribution of MSE between the AE predictions and the input strain for two types of studied signals: injected GW (red histograms) and noise (blue histograms). Plots correspond to the $L1$ (*left plot*), $H1$ (*middle plot*) and $V1$ (*right plot*) datasets. Additional black vertical line representing anomaly DT was added to emphasize amount of detected anomalies (hatched areas).

reconstructed mainly the merger part. Overall, the AE seemed to fail to reconstruct lower amplitudes and frequencies of the GW signal (the inspiral and ringdown part).

To compute the threshold for the anomaly detection, we generated histograms of the MSE for every detectors' dataset and compared its values with the FPR. The results are shown in figure 12. The anomalies covered a greater range of MSE values than the noise, with an overlapping range varying for different datasets (coloured in burgundy in figure 12). In the case of $L1$ data, this region was the smallest whereas for $V1$, it was the largest. As in the case of simulated data, we defined a detection threshold for anomalies assuming FPR = 5%, which resulted in the following thresholds: $\mathrm{DT}_{L1} = 1.3 \times 10^{-5}$ for $L1$, $\mathrm{DT}_{H1} = 2.2 \times 10^{-5}$ for $H1$ and $\mathrm{DT}_{V1} = 4.3 \times 10^{-6}$ for $V1$. The results of the anomaly searches on real datasets are present in table 2 in the form of confusion matrix. Anomaly row relates to the hatched area on panels in figure 12. In the case of $L1$ and $H1$ datasets, around half of detected anomalies are correctly related to the injected GW. Samples that did not exceed corresponding DT had low SNR. That was also the case of the $V1$ dataset where only 27% of samples exceeded DT_{V1}. Details presenting the relation between the SNR of the injected GW and MSE of the reconstructed waveform for the real datasets can be found in the appendix B.

Table 2. Results of anomaly detection of CNN-AE at FPR = 5% for $L1$, $H1$ and $V1$ datasets in the form of confusion matrix. Columns relate to the ground-truth values whereas rows to the predictions. For all datasets significant majority of detected anomalies correctly corresponded to the data instances with injected GW. However, more than a third part of non-anomalous class (samples that did not exceed DT for a given detector) related to the low SNR injected GW (see figure 6 for comparison between $L1$, $H1$ and $V1$ GW SNR distributions).

	$L1$		$H1$		$V1$	
	Inj. GW	Noise	Inj. GW	Noise	Inj. GW	Noise
Anomaly	52%	5%	50%	5%	27%	5%
Non-anomaly	48%	95%	50%	95%	73%	95%

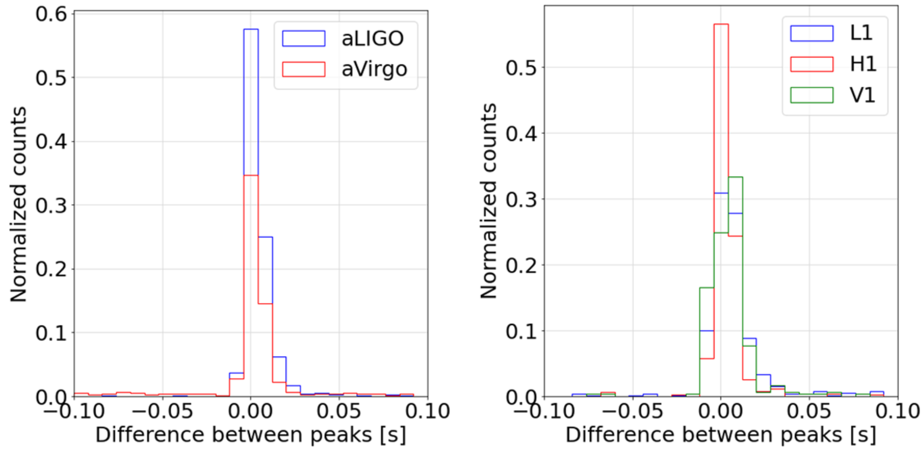

Figure 13. Anomaly localisation based on the difference in peak positions between reconstructed and injected signals. *Left plot*: The results for the simulated datasets. *Right plot*: The results for the real datasets.

Additionally, we performed tests on the localisation in time of anomalies detected by our method. We compared the known times of the GW injection into the data with the time of the reconstructed signal. Specifically, we subtracted the times corresponding to the maximum amplitude peaks of both signals, and plotted histograms of the resulting differences for all the anomalies that exceeded computed previously DT. For comparison, we performed this procedure also for the simulated datasets. The results are shown in figure 13. In all the studied cases, around 95% of detected anomalies were localised within 0.05 s intervals around the injection times. We conclude that this feature may be useful not only for detection and reconstruction, but also potentially for other applications, such as the localisation of signals in the data from many detectors, and subsequent sky localisation of the sources.

4.3. Anomaly searches on confirmed GW detections

Using a selection of real GW detections provided by the LIGO-Virgo collaboration on the GWOSC platform [40], we tested the AE on three relatively strong signals from the GWTC-1 O1-O2 catalog [39]: GW150914 [1], GW170608 [48] and GW170814 [49]. The reported network SNR (square root of a sum of squares of SNRs from individual detectors ρ_i) $\rho_{\mathrm{net}} = \sqrt{\sum_i \rho_i^2}$ is $\simeq$24 for GW150914, $\simeq$15 for GW170608 and $\simeq$16 for GW170814 [40]. Assuming two equally sensitive detectors, each of them would measure SNR of $\simeq$17 for GW150914, $\simeq$10 for GW170608 and and $\simeq$11 for GW170814; note that the GW170814 was a three-detector event, with the single-detector SNRs in $H1$, $L1$ and $V1$ equal to 7.3, 13.7, and 4.4, respectively [49]. In reality, due to differences in sensitivity the detectors registered the signals with different SNRs, which were nevertheless near the single-detector SNR detection threshold, established by the FPR = 5% condition.

After whitening, test data were fed into the AE. Then, the reconstructed values were subtracted from the input data. The results of subtraction for GW150914 are shown on the two upper, right plots in figure 14 for the both LIGO detectors. The presented signals resulted in the $\mathrm{MSE}_{L1} = 3.2 \times 10^{-4}$ and $\mathrm{MSE}_{H1} = 1.0 \times 10^{-3}$. For both detectors, MSE had significantly higher values than the corresponding detection thresholds of FPR = 5%.

In case of GW170608, the AE detected the event; however, the reconstructed signal was limited to the merger part for both the LIGO detectors as shown on the two middle, right plots in figure 14. A potential explanation for this weaker reconstruction of this particular GW is related to the different mass ranges of GW170608 and the GWs used for the training of the AE. GW170608 had BBH component masses

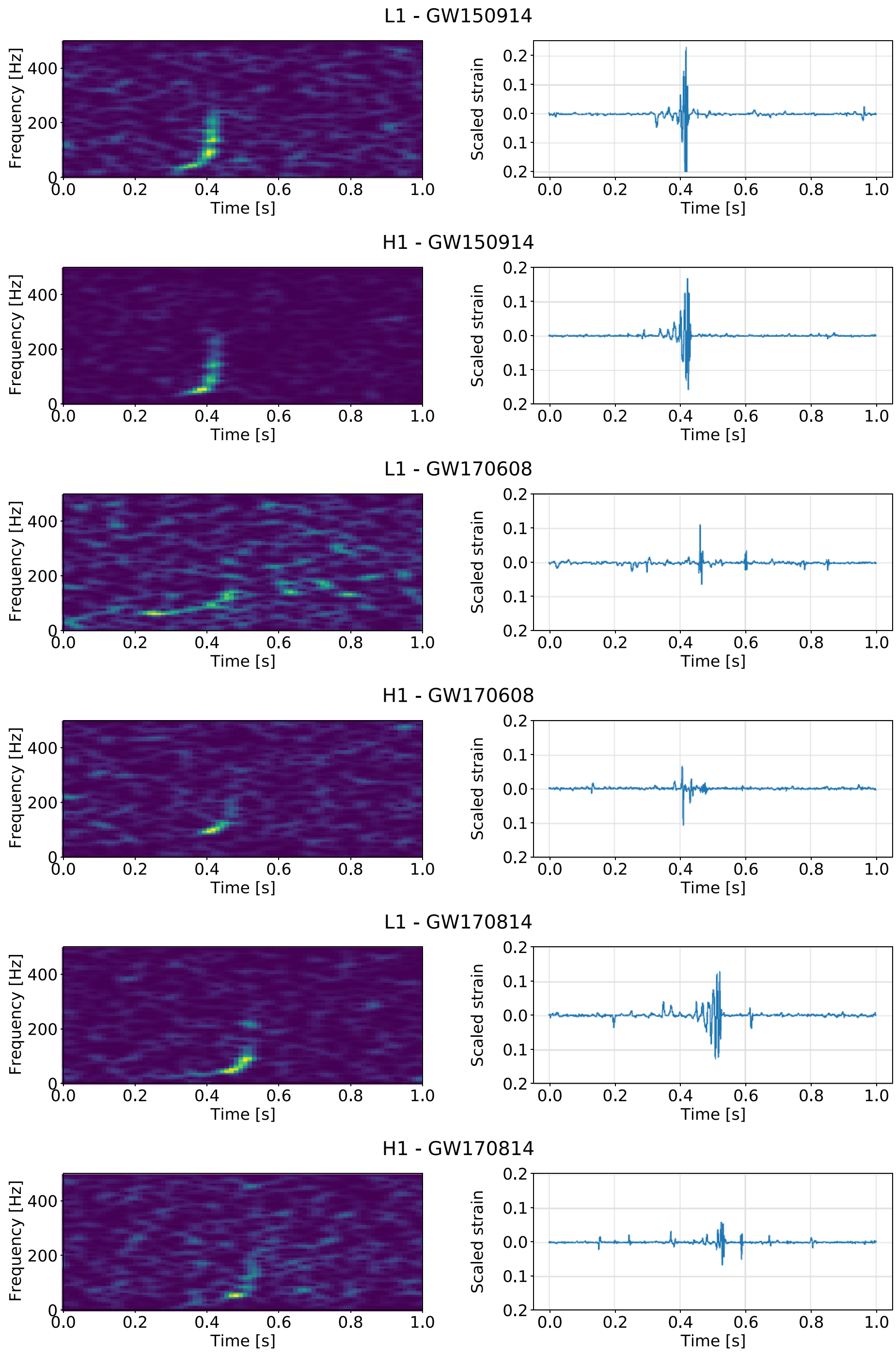

Figure 14. Test of the CNN-AE on the dataset containing confirmed detections using $L1$ and $H1$ datasets for GW150914 (four upper plots), GW170608 (four middle plots) and GW170814 (four bottom plots). *Left plots*: spectrograms presenting the relation between frequency and time for the segment of real data containing GW. *Right plots*: the difference between the CNN-AE predictions and the input data.

corresponding to around $m_1 = 11.0$ and $m_2 = 7.6\ M_\odot$ which were substantially smaller than the masses used in the data generation (see section 3 for more details). We note that the H1 detector was nominally outside of observing mode at the time of this event. This data release includes H1 data around the time of this event, by using a modified segment list, as was done for the published analysis [39, 40, 48]. Nevertheless, the AE detected GW170608, proving its generalisation capabilities towards recognising gravitational waveforms it was not trained for. This is a sure advantage of our proposed method, when compared to the matched filtering method. Furthermore, the MSE values for both detectors had slightly higher values above the detection thresholds at FPR = 5%: $\mathrm{MSE}_{L1} = 5.3 \times 10^{-5}$ and $\mathrm{MSE}_{H1} = 4.1 \times 10^{-5}$.

The last studied test case, GW170814, was detected in both LIGO detectors with a substantial part of the waveform being recovered, as shown on the two bottom, right plots in figure 14. For the $V1$ data, AE failed to detect the event evidently due to low reported SNR of $\simeq$4 [49]. The MSE for the $L1$ and $H1$ datasets were above the data threshold, at 5% and equal to $\mathrm{MSE}_{L1} = 2.2 \times 10^{-4}$, $\mathrm{MSE}_{H1} = 2.2 \times 10^{-4}$, whereas for $V1$ the MSE was below the threshold: $\mathrm{MSE}_{V1} = 1.8 \times 10^{-6}$.

These results confirm that our proposed method of using CNN-AE for anomaly searches is able to detect real GWs, even though the deep learning model was trained on relatively uncomplicated datasets, based on the information related to a particular GW waveform models, varying with respect to the limited range of masses and distances.

5. Summary

In this paper, we proved that AEs are a potentially powerful method for anomaly searches in GW data. A relatively simple AE, consisting of only three hidden layers, was capable of detecting anomalies, defined in terms of transient BBH GWs (as well glitches in real data), and trained on either simulated or real data. Moreover, our proposed method was able to detect as anomalies all three confirmed GWs used as a test case and even partially reconstruct the waveforms of the underlying signals.

In the proposed method we introduced a metric allowing for the automatic detection of anomalies. The chosen metric was MSE. Using this metric, we defined the threshold for the anomaly detection by associating MSE with FPR. The results presented in the manuscript referred to FPR = 5%. At such a threshold we were able to detect from the LIGO data set, almost all of the injected GW into the simulated dataset, as well as around 50% for real detector data. In the case of Virgo, for simulated data around half of injected signals were detected and quarter for real data. The reason for worse results on real data taken from the O1 to O2 observational run, was the substantial difference in the sensitivity with respect to the simulated counterpart. The sensitivity of real detectors were overall worse than the designed one. However, the real sensitivity significantly improved for the O3 observational run after detector upgrades. Once the data from O3 becomes public, the anomaly searches of the proposed CNN-AE method are expected to improve.

Our method also proved to be useful in the precise time localisation of anomalies. Anomalies exceeding the detection threshold for all the studied datasets (real and simulated) turned out to be localised with accuracy below 0.05 s. Localisation of the peak amplitude may be particularly useful for the localisation of GW sources on the sky in multi-detector analysis. However, such an application of our method requires a separate study.

Finally, the successful detection of GW170608 proved the generalisation capabilities of our AE, towards the detection of GWs with parameters that are different than those of the GWs used for the AE training and in the data nominally outside of the observing mode.

Among the future projects we are considering are applications of recurrent neural networks instead of CNN in the anomaly searches as an alternative suited for the time-series data as well as anomaly searches of different GW types such as core collapse supernova signals.

Data availability statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments

The work was partially supported by the Polish NCN Grant Nos. 2016/22/E/ST9/00037, 2017/26/M/ST9/00978 and 2020/37/N/ST9/02151, as well as the European Cooperation in Science and Technology COST action G2Net (CA17137). The Quadro P6000 used in this research was donated by the NVIDIA Corporation. The production computations were supported in part by PL-Grid Infrastructure and by the MNiSW grant for expansion of the strategic IT infrastructure for science.

This research has made use of data, software and/or web tools obtained from the Gravitational Wave Open Science Center (www.gw-openscience.org), a service of LIGO Laboratory, the LIGO Scientific Collaboration and the Virgo Collaboration. LIGO Laboratory and Advanced LIGO are funded by the United States National Science Foundation (NSF) as well as the Science and Technology Facilities Council (STFC) of the United Kingdom, the Max-Planck-Society (MPS), and the State of Niedersachsen/Germany for support of the construction of Advanced LIGO and construction and operation of the GEO600 detector. Additional

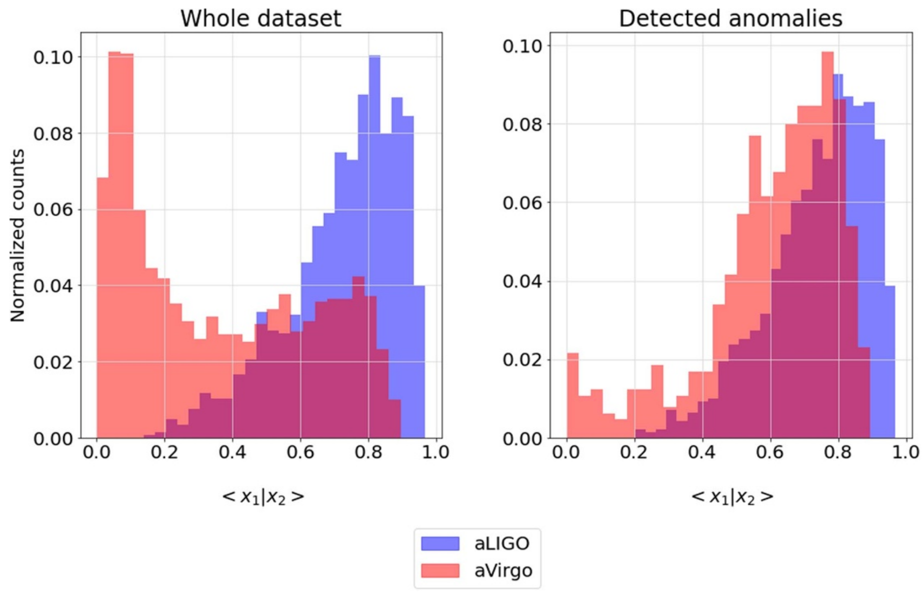

Figure 15. Distribution of a match between the injected (x_1) and the reconstructed (x_2) waveforms as a function of normalized scalar product $\langle x_1|x_2\rangle$. Scalar product equals to one relates to the full match between waveforms whereas value of zero—no match. *Left plot*: results for the whole studied dataset of particular detector; *right plot*: results of a match after applying the anomaly detection threshold at FPR = 5%.

support for Advanced LIGO was provided by the Australian Research Council. Virgo is funded, through the European Gravitational Observatory (EGO), by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale di Fisica Nucleare (INFN) and the Dutch Nikhef, with contributions by institutions from Belgium, Germany, Greece, Hungary, Ireland, Japan, Monaco, Poland, Portugal, Spain.

Appendix A. Match/overlap between the injected and reconstructed waveforms

A.1. Simulated dataset

To measure the match between the injected and the reconstructed waveforms we used the normalized scalar product $\langle x_1|x_2\rangle$ in the time domain resulting in values in a range $(0,1)$. Zero related to no match, whereas one to full match between the waveforms. Presented results in figure 15 corresponds to the whole studied dataset for aLIGO and aVirgo detectors (left panel) as well as samples exceeding the anomaly detection thresholds (right panel). Applying respective DT allowed to substantially reduce number of samples with no match between the waveforms as a result of low SNR of injected GW. Samples exceeding DT were reconstructed to a greater extent which resulted in $\langle x_1|x_2\rangle$ closer to one.

A.2. Real dataset

The same metric as in case of simulated dataset was used to study the match between the injected and the reconstructed waveforms for the real datasets. Presented in figure 16 results show the similarities in the match between consecutive datasets ($L1$, $H1$ and $V1$) as well as the effect of applying the anomaly detection threshold. Samples with $\langle x_1|x_2\rangle$ close to zero had low SNR. As a result they were poorly reconstructed which in turn translated into low value of MSE. Samples exceeding respective DT were reconstructed to a greater extent as in the case of simulated data. However, overall match was worse—the mean values of $\langle x_1|x_2\rangle$ for real datasets were around 0.6, whereas for simulated data 0.8.

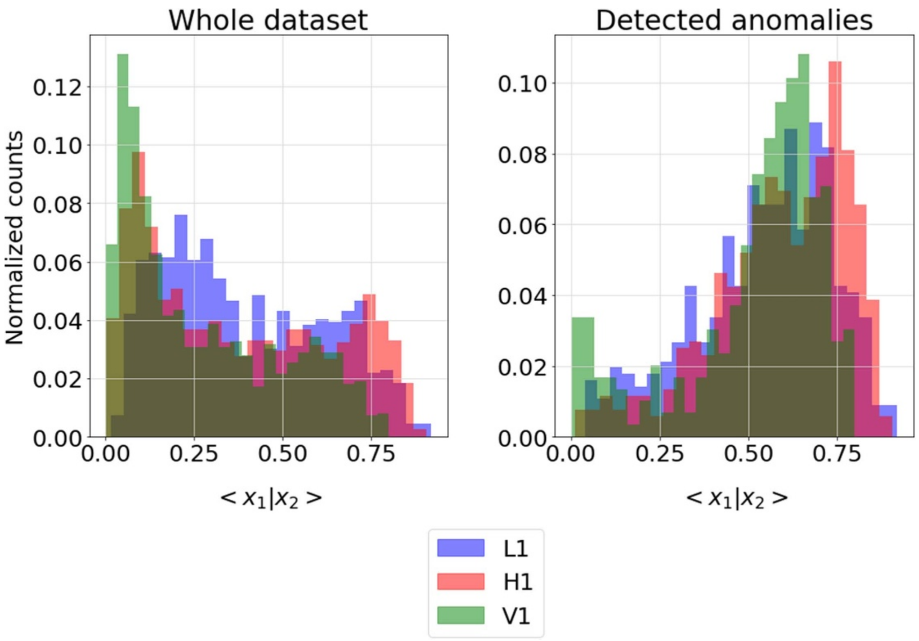

Figure 16. Distribution of a match between the injected (x_1) and the reconstructed (x_2) waveforms as a function of normalized scalar product $\langle x_1|x_2\rangle$. Scalar product equals to one relates to the full match between waveforms whereas value of 0—no match. *Left plot*: results for the whole studied dataset of particular detector; *right plot*: results of a match after applying the anomaly detection threshold at FPR = 5%.

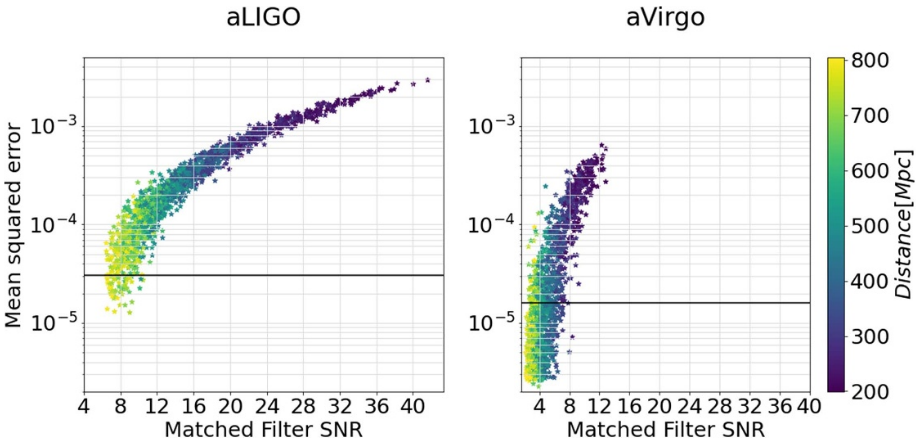

Figure 17. Relation between MSE and matched filter SNR for aVirgo and aLIGO datasets colored with a distance to the GW source. Black horizontal line corresponds to the detection threshold of anomalies at FPR = 5%.

Appendix B. Signal-to-noise ratio vs mean squared error

B.1. Simulated dataset

For the aLIGO dataset above SNR = 20, the MSE-SNR relation was almost linear, with a small spread of individual data instances along MSE, as shown in figure 17. Whereas with the decline of SNR, the spread of MSE significantly increased, characterized by the non-linearity in the MSE-SNR relation. Anomalies around the same SNR for values below ten varied up to an order of magnitude in MSE. Manual inspection of data samples containing anomalies of low SNR provided an explanation of this behaviour. In the analysed samples, only the merger part of the gravitational waveform was recovered. For lower SNRs (below ten) the recovery was partial and dependent on the variability of the noise. If the amplitude of the noise in a given data segment was small comparing to the injected GW signal, the resulting MSE was higher than in the case of noise samples with larger amplitudes. Overall, for the aLIGO dataset, the susceptibility of AE to the local variability of the noise was inversely proportional to the SNR of the injected anomaly.

In case of the aVirgo dataset, the mentioned susceptibility was more significant. Matched filter SNRs for all injected GWs covered a smaller range of values than for aLIGO (compare SNR ranges on the bottom right

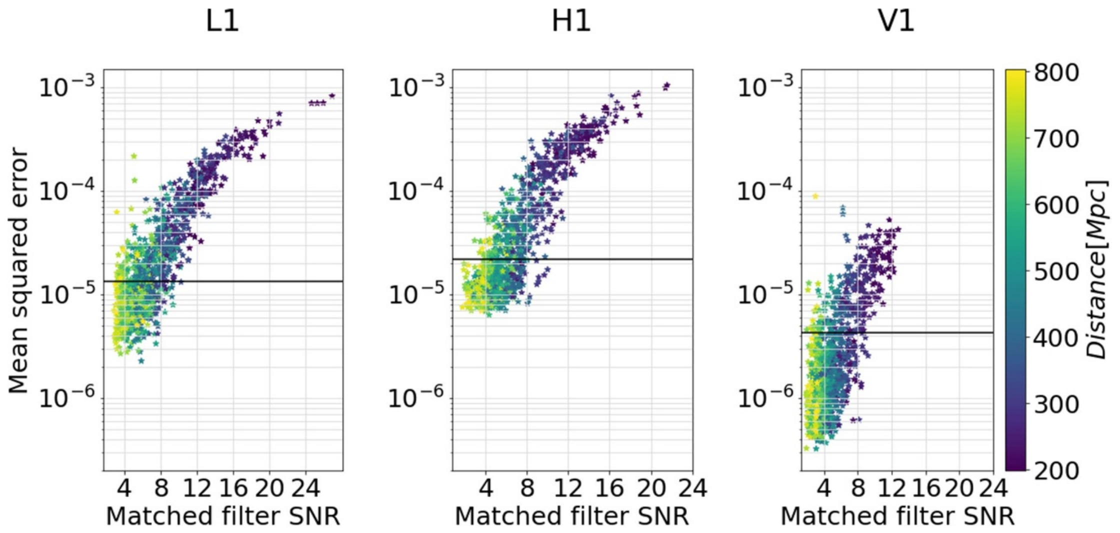

Figure 18. Relation between the MSE and the matched filter SNR for $L1$ (*left plot*), $H1$ (*middle plot*) and $V1$ (*right plot*) datasets colored with a distance to the GW source. Black horizontal line corresponds to the detection threshold of anomalies at FPR = 5%.

plot in figure 6 for aLIGO and aVirgo datasets). In the majority of studied cases, the recovery of the anomaly was partial and limited to the merger part.

B.2. Real dataset

The relation between SNR and MSE presented similar features as for the simulated data discussed above. The variability of MSE for samples of similar SNR was largest among the weakest anomalies as shown in figure 18. The increase of the matched filter SNR led to the decrease of the spread in MSE, and thus their relation became more linear. This was the case for the AEs trained on the $L1$ and $H1$ datasets. In contrast, the AE trained on the $V1$ dataset was more susceptible to the local variability of the noise. Since the injected anomalies had a small SNR for $V1$ (majority of injected GW had SNR below ten), their amplitude was significantly lower than the detectors noise. As a result, the anomaly detection depended on the variability of the noise, which had a random character.

ORCID iDs

Filip Morawski https://orcid.org/0000-0002-6194-8239
Michał Bejger https://orcid.org/0000-0002-4991-8213
Elena Cuoco https://orcid.org/0000-0002-6528-3449
Luigia Petre https://orcid.org/0000-0002-0648-3301

References

[1] Abbott B P *et al* 2016 *Phys. Rev. Lett.* **116** 061102
[2] Aasi J *et al* 2015 *Class. Quantum Grav.* **32** 074001
[3] Acernese F *et al* 2015 *Class. Quantum Grav.* **32** 024001
[4] LIGO Scientific Collaboration 2020 O3 summary (available at: www.ligo.caltech.edu/WA/news/ligo20200326)
[5] Virgo Collaboration 2020 Virgo status (available at: www.virgo-gw.eu/status.html)
[6] LIGO Scientific Collaboration and Virgo Collaboration 2020 Gravitational-wave candidate event database (available at: https://gracedb.ligo.org/superevents/public/O3/)
[7] Abbott R *et al* 2020 GWTC-2: compact binary coalescences observed by LIGO and virgo during the first half of the third observing run (arXiv:2010.14527)
[8] Abbott B P *et al* 2019 *Astrophys. J. Lett.* **882** L24
[9] Acernese F *et al* 2016 *Phys. Rev.* D **93** 122003
[10] Kumar A, Saminadayar L, Glattli D C, Jin Y and Etienne B 1996 *Phys. Rev. Lett.* **76** 2778–81
[11] Owen B and Sathyaprakash B 1999 *Phys. Rev.* D **60** 022002
[12] Usman S A *et al* 2016 *Class. Quantum Grav.* **33** 215004
[13] Sachdev S *et al* 2019 (arXiv:1901.08580)
[14] Klimenko S *et al* 2016 *Phys. Rev.* D **93** 042004
[15] Klimenko S *et al* 2008 *Class. Quantum Grav.* **25** 114029
[16] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT Press)
[17] Samuel A L 1959 *IBM J. Res. Dev.* **3** 210–29
[18] Robinet F, Arnaud N, Leroy N, Lundgren A, Macleod D and McIver J 2020 *SoftwareX* **12** 100620

[19] Chatterji S, Blackburn L, Martin G and Katsavounidis E 2004 *Class. Quantum Grav.* **21** S1809–18

[20] George D and Huerta E 2018 *Phys. Lett.* B **778** 64–70

[21] Shen H, George D, Huerta E A and Zhao Z 2019 Denoising gravitational waves with enhanced deep recurrent denoising auto-encoders *ICASSP 2019—2019 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* pp 3237–41

[22] Dreissigacker C, Sharma R, Messenger C, Zhao R and Prix R 2019 *Phys. Rev.* D **100** 044009

[23] Morawski F, Bejger M and Ciecielag P 2020 *Machine Learning: Science and Technology* **1** 025016

[24] Beheshtipour B and Papa M A 2020 *Phys. Rev.* D **101** 064009

[25] Razzano M and Cuoco E 2018 *Class. Quantum Grav.* **35** 095016

[26] Iess A, Cuoco E, Morawski F and Powell J 2020 *Machine Learning: Science and Technology* **1** 025014

[27] Corizzo R, Ceci M, Zdravevski E and Japkowicz N 2020 *Expert Syst. Appl.* **151** 113378

[28] Giles D and Walkowicz L 2019 *MNRAS* **484** 834–49

[29] D'Addona M, Riccio G, Cavuoti S, Tortora C and Brescia M 2020 Anomaly detection in astrophysics: a comparison between unsupervised deep and machine learning on kids data (arXiv:2006.08235)

[30] Baron D 2019 (arxiv:1904.07248)

[31] Farina M, Nakai Y and Shih D 2020 *Phys. Rev.* D **101** 075021

[32] Baldi P 2011 Autoencoders, unsupervised learning and deep architectures *Proc. 2011 Int. Conf. on Unsupervised and Transfer Learning Workshop (UTLW'11)* vol 27 (JMLR.org) pp 37–50

[33] Kingma D P and Ba J 2014 (arxiv:1412.6980)

[34] Van Rossum G and Drake F L 2009 *Python 3 Reference Manual* (Scotts Valley, CA: CreateSpace)

[35] Chollet F *et al* 2015 Keras (available at: https://keras.io)

[36] Abadi M *et al* 2015 TensorFlow: large-scale machine learning on heterogeneous systems software available from tensorflow.org (available at: www.tensorflow.org)

[37] Nickolls J, Buck I, Garland M and Skadron K 2008 *Queue* **6** 40–53

[38] Chetlur S, Woolley C, Vandermersch P, Cohen J, Tran J, Catanzaro B and Shelhamer E 2014 (arxiv:1410.0759)

[39] Abbott B P *et al* 2019 *Phys. Rev.* X **9** 031040

[40] Abbott R *et al* 2019 arxiv:1912.11716

[41] Cuoco E, Calamai G, Fabbroni L, Losurdo G, Mazzoni M, Stanga R and Vetrano F 2001 *Class. Quantum Grav.* **18** 1727–51

[42] Nitz A *et al* 2020 gwastro/pycbc: Pycbc release v1.16.10 (available at: https://doi.org/10.5281/zenodo.4063644)

[43] Hannam M, Schmidt P, Bohé A, Haegel L, Husa S, Ohme F, Pratten G and Pürrer M 2014 *Phys. Rev. Lett.* **113** 151101

[44] Salpeter E E 1955 *Astrophys. J.* **121** 161

[45] The Virgo Collaboration 2009 *Advanced Virgo Baseline Design VIR-027A-09* (available at: https://tds.virgo-gw.eu/?call_file=VIR-0027A-09.pdf)

[46] Virgo Collaboration 2020 Virgo interferometer monitoring webpage (available at: https://vim-online.virgo-gw.eu)

[47] LIGO Scientific Collaboration 2020 Updated advanced LIGO sensitivity design curve (available at: https://dcc.ligo.org/LIGO-T1800044/public)

[48] Abbott B P *et al* 2017 *Astrophys. J.* **851** L35

[49] Abbott B P *et al* (LIGO Scientific Collaboration and Virgo Collaboration) 2017 *Phys. Rev. Lett.* **119** 141101